FRONTIER ENERGY SYSTEM AND POWER ENGINEERING
ISSN: 2720-9598

Forecasting Hourly Energy Fluctuations Using Recurrent Neural Network (RNN)

Aji Prasetya Wibawa [1], Ade Kurnia Ganesh Akbari [2], Akhmad Fanny Fadhilla [3], Alfiansyah Putra Pertama Triono [4], Andien Khansa'a Iffat Paramarta [5], Faradini Usha Setyaputri [6], Agung Bella Putra Utama [7], Jehad A.H. Hammad [8]

[1,2,3,4,5,6,7] Department of Electrical Engineering and Informatics, Faculty of Engineering, Universitas Negeri Malang, Jl. Semarang no. 5, Malang 65145, Indonesia
[8] Department of Computer Information Systems, Al-Quds Open University, Beit Jalla, The Main road - Khallat Al Badd, Bethlehem, Palestine
Corresponding: aji.prasetya.ft@um.ac.id

Abstract

Energy forecasting is currently essential due to its various benefits. Energy data analysis for forecasting requires a functional method due to the complexity of the observed data. This forecasting study used the Recurrent Neural Networks (RNN) method. Parameters used include batch size, epoch, hidden layers, loss function, and optimizer obtained from hyperparameter tuning grid search. A comparison of different normalization methods, namely min-max, and z-score, was conducted. Using min-max normalization yielded the best performance with MAPE of 3.9398%, RMSE of 0.0630, and R2 of 0.8996. In testing with z-score normalization, it showed a performance of MAPE of 10.6120%, RMSE of 0.7648, and R2 of 0.4142.

Keywords

Energy Forecasting, RNN, Normalization, Grid Search

INTRODUCTION

With the advancement of time and the growth of the population, the energy demand has steadily increased. However, the availability of energy remains limited. Therefore, research is needed to balance energy demand and supply. Energy data is complex due to its temporal nature, ranging from years, months, days, and even hours.

Time series data involves a sequence of time-related observations that influence each other [1]. Time series analysis examines and analyzes time series data [2]. Based on the number of attributes, time series can be categorized into two types: univariate time series, which has only one attribute [3], and multivariate time series, which has more than one attribute that interacts with each other [4]. Multivariate time series analysis has been applied in various fields such as healthcare [5], finance [6], geography [7], and natural sciences [8]. This demonstrates that each attribute tends to depend on changes in other attributes [9].

From the attributes present in time series data, it is often the case that these data have different ranges [10]. Therefore, a function is needed to address this issue. This technique is called normalization [11], a function that transforms the values of each data point to be consistent and processable by deep learning models. Deep learning is a subset of machine learning that employs a system modeled after the human brain's neural network to analyze data [12]. Several deep learning methods that can be applied in the analysis process include Recurrent Neural Network (RNN) [13], Convolutional Neural Network (CNN) [14], Long Short-term Memory (LSTM) [15], Gated Recurrent Unit (GRU) [16], and Bidirectional LSTM (Bi-LSTM) [17]. RNN is the most fundamental method in developing time series data forecasting due to its recurrent process.

Based on the previous researchers' descriptions, this study applies multivariate time series data analysis using the RNN method. All attributes in the data will be used with multiple scenarios to achieve the best accuracy. Performance evaluation will be measured using the values of Mean Absolute Percentage Error (MAPE), Root Mean Square Error (RMSE), and R-squared (R2).

METHOD

This research is conducted by following the steps outlined in CRISP-DM. CRISP-DM is the industry's most commonly used method in data mining [18]. Figure 1 illustrates the CRISP-DM process flow.

FRONTIER ENERGY SYSTEM AND POWER ENGINEERING
ISSN: 2720-9598

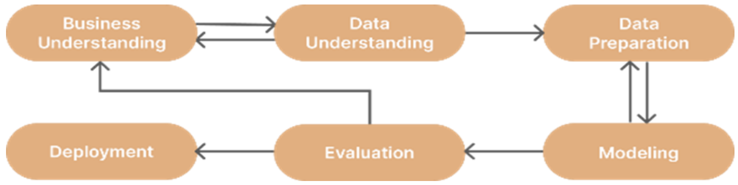


Figure 1. CRISP-DM Process Flow

1. Business Understanding

In multivariate time series analysis using deep learning models, it is essential to consider the data's condition. Variations in the range of values in multivariate time series data can negatively impact the analysis's performance. Data normalization is required to standardize the value ranges to address this issue. There are several normalization methods available, such as min-max and z-score. The selection of the appropriate normalization method can significantly influence the accuracy of the analysis process.

2. Data Understanding

The data used for this research spans from January 2015 to December 2018. The dataset file contains 35,064 instances with 29 attributes. Details of the dataset can be found in Table 1.

TABLE 1

DESCRIPTION DATA

Attribute	Data type	Description (min, max)
Generation biomass	Float	(0, 592)
Generation of fossil brown coal/lignite	Float	(0, 999)
Generation of fossil coal/derived gas	Float	(0)
Generation fossil gas	Float	(0, 20034)
Generation fossil hard coal	Float	(0, 8359)
Generation fossil oil	Float	(0, 449)
Generation fossil oil shale	Float	(0)
Generation fossil peat	Float	(0)
Generation geothermal	Float	(0)
Generation hydro-pumped storage aggregated.	Float	NaN
Generation hydro-pumped storage consumption.	Float	(0, 4523)
Generation hydro run-off river and poundage	Float	(0, 2000)
Generation hydro water reservoir	Float	(0, 9728)
Generation marine	Float	(0)
Generation Nuclear	Float	(0, 7117)
Generation other	Float	(0, 106)
Generation other renewable	Float	(0, 119)
Generation solar	Float	(0, 5792)
Generation waste	Float	(0, 357)
Generation wind offshore	Float	(0)
Generation wind onshore	Float	(0, 17436)
Forecast solar day.	Float	(0, 5836)
Forecast wind offshore day ahead	Float	NaN
Forecast wind onshore day ahead	Float	(0, 17430)
Total load forecast	Float	(0, 41390)
Total load actual	Float	(0, 41015)
Price day ahead	Float	(0, 98.69)
Price actual	Float	(0, 99.95)

The target attribute is "Total load actual," which contains the energy demand values. However, the "Total load forecast" attribute is not utilized, resulting in 28 attributes being used.

3. Data Preparation

In this study, data preparation involves a preprocessing stage to make the data acceptable to the model and to facilitate the testing process. The preprocessing step conducted in this research is handling missing values. The discovery of missing values in data for analysis is a common issue and can potentially affect model performance. Several techniques are available to handle missing values, and one of the methods employed in this study is deletion. This technique involves removing rows or columns containing missing values.

ISSN: 2720-9598

4. Modeling

The forecasting model using the RNN method employs either min-max or z-score normalization. Further details of the forecasting model can be found in Figure 2.

From Figure 2, in the data preprocessing stage, data normalization is a technique in data preprocessing used to adjust the values of each data point to the same range. With this technique, each variable in the data will have an equal impact on the model's performance. This means that none of the variables will have an excessively large influence. The choice of normalization method can affect the accuracy of an analysis. In this study, the implemented methods are:

- Min-max normalization is a method that transforms the values in the data into a range of 0 to 1. The minimum value becomes 0, the maximum value becomes 1, and the values in between will be decimal values between 0 and 1.
- Z-Score normalization, commonly known as standardization, transforms the values in the data into values ranging from -1 to 1. This technique is based on measurements using the mean and standard deviation.

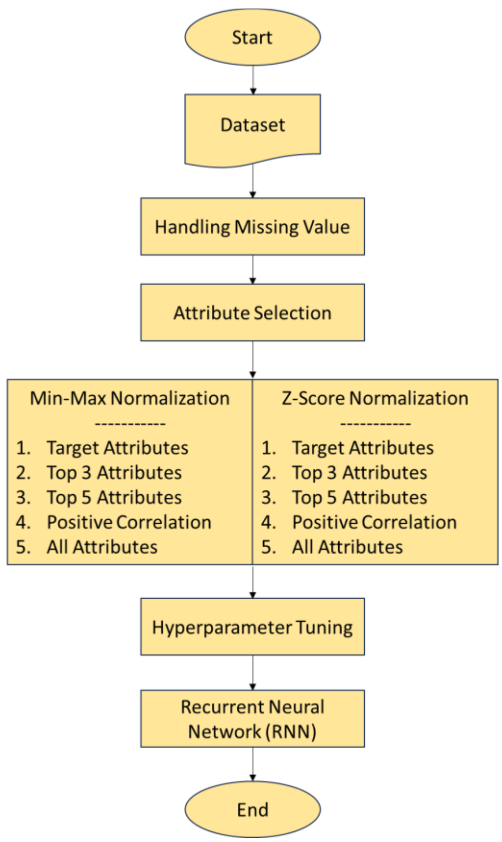


Figure 2. Forecasting Model

After the preprocessing stage, testing is conducted using the Recurrent Neural Networks (RNN) Method. The data is split into a training and testing ratio 100:100 to maximize data usage. The other parameters used are determined by the grid search process [19]. Batch sizes of 100 and 1000 are selected based on the number of data instances used. The choice of batch size also impacts computational load, with smaller batch sizes leading to reduced computational load [20]. Similar to batch size, the selection of the number of epochs depends on the number of instances. Too few epochs can hinder the model from converging. Adam and Rmsprop are chosen as options for optimizers within the search space. This choice is due to RMSprop's evolution of Adagrad, while Adam combines RMSprop with momentum. Both optimizers yield faster convergence than other optimizers [21]. MSE, MAE, and Huber loss are regression loss functions.

5. Evaluation

The evaluation of error values in this study employs the methods of MAPE (Mean Absolute Percentage Error), RMSE (Root Mean Square Error), and R-squared (R2). By calculating these errors and accuracy measures, it will be possible to determine which model performs best in forecasting.

6. Deployment

In this stage, the evaluation results are reported by comparing two normalization methods, min-max, and z-score, during the testing process of the RNN model. The analysis compares the average values of MAPE, RMSE, and R2. The normalization method that yields the lowest MAPE and RMSE values and the highest R2 value will be considered the best normalization method in this study.

RESULT

This section discusses the results and discussion of the research using CRISP-DM. It is systematically organized into several sections: Data Preparation, Modeling, Evaluation, and Deployment. Meanwhile, the business understanding and data understanding stages are discussed in detail in Method.

1. Data Preparation Results

In the data preparation stage, the dataset's missing values are handled by removing columns with attributes with NaN values. Among the 28 attributes used in the dataset, there are two attributes with NaN values: "Generation hydro pumped storage aggregated" and "Forecast wind offshore day ahead." After removal, the number of attributes is reduced from 28 to 26.

2. Modeling Result

The modeling process begins with calculating the correlation of each attribute with the target attribute, which is the "Total load actual." The correlation calculation is assisted using a Python library. After determining the correlation values, the attributes are ranked according to the attribute selection scenario, excluding attributes with negative values and low correlation values (NaN). Normalization is carried out after the attribute selection stage. Normalization serves to standardize different value ranges into a specific range. This ensures that deep learning models can accept multivariate time series data in the analysis. In this stage, min-max and z-score normalization are performed using Python programming libraries. Min-max normalization is used to transform values in the data into the same range, which is between 0 and 1. Meanwhile, z-score normalization is used to transform values in the data into the range of -1 to 1. The results of min-max and z-score normalization for one of the attributes, "Total load actual," can be seen in Table 2.

TABLE 2

NORMALIZATION RESULTS

Actual Value	Min-max Normalization	Z-Score Normalization
25385	0.319666	-0.723934
24382	0.276008	-0.943172
⋮	⋮	⋮
29735	0.509010	0.226902
28071	0.436580	-0.136820

The final stage of modeling is Hyperparameter Tuning. In this stage, grid search is employed to find the parameters that yield optimal performance. The hyperparameter tuning results consider using the tanh activation function centered at 0 and a dropout rate of 0.2 to prevent overfitting. The hyperparameter tuning in this research produces results, as shown in Table 3.

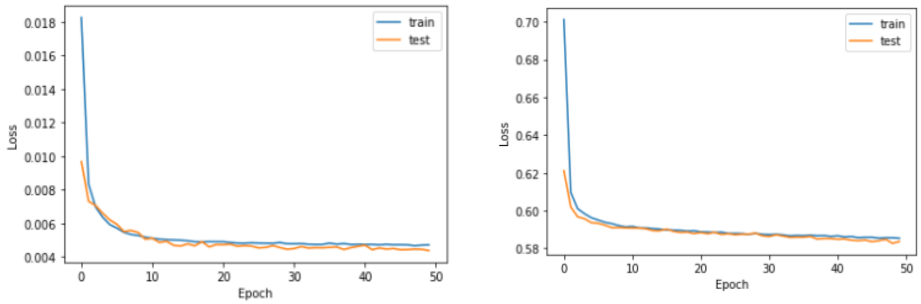
TABLE 3

HYPERPARAMETER TUNING RESULTS

Parameter	Search Space	Result
Batch Size	‘100’, ‘1000’	100
Epoch	’50’, ‘100’	50
Hidden Layer	‘2’, ‘5’, ‘10’	2
Loss Function	‘mse’, ‘mae’, ’huberloss’	MSE
Neuron	‘32’, ‘64’	32
Optimizer	‘adam’, ‘rmsprop’	Rmsprop

ISSN: 2720-9598

From Table 3, it is explained that using the parameters obtained from grid search results in a converging loss value at epoch 50. This indicates that the model has stopped learning since epoch 50 and reached its highest capability. The convergence value of min-max and z-score normalization is presented in Figure 3.



(a) (b)

Figure 3. Convergence chart, (a). Min Max, (b). Z-Score

3. Evaluation Results

The evaluation results are obtained by calculating the MAPE, RMSE, and R2 average using five experiments. The scenario with the best results is characterized by the lowest MAPE and RMSE values and the highest R2 value. Table 4 and Table 5 illustrate the experiment results using min-max and z-score normalization across five attribute selection scenarios. Each scenario shows insignificant differences in the resulting MAPE, RMSE, and R2 values. The scenario with the top 5 attributes yields the best MAPE, the scenario with all attributes without NaN and negative attributes yields the best RMSE, while the scenario with all attributes provides the best R2.

TABLE 4

MIN-MAX NORMALIZATION PERFORMANCE RESULTS

Scenario	MAPE (%)	RMSE	R^2
Target Attributes	3.9398	0.0630	0.8996
Top 3 Attributes	3.9821	0.0630	0.8995
Top 5 Attributes	3.9622	0.0631	0.8993
Positive correlation	4.3125	0.0659	0.8902
All Attributes	4.2951	0.0656	0.8911

Table 4 explains the performance results with min-max normalization. The best MAPE is achieved by the target attribute scenario with a value of 3.9398%, and the best RMSE is obtained from the target attribute and the top 3 attribute scenario with a value of 0.0630, and the best R2 is found in the target attribute with a value of 0.8996.

TABLE 5

Z-SCORE NORMALIZATION PERFORMANCE RESULTS

Scenario	MAPE (%)	RMSE	R^2
Target Attributes	10.6537	0.7663	0.4124
Top 3 Attributes	10.6985	0.7670	0.4110
Top 5 Attributes	10.6543	0.7655	0.4133
Positive correlation	10.6120	0.7648	0.4142
All Attributes	10.6165	0.7649	0.4142

Based on the results outlined in Table 5, the scenario with the best MAPE, RMSE, and R2 values is the one without NaN and negative attributes, with values of 10.6120%, 0.7648, and 0.4142, respectively. However, the R2 value for all attributes also provides the best result with the same values as without NaN and negative attributes.

3.4 Deployment Results

The Deployment phase is carried out to report the evaluation results between the two normalization methods in the testing process of the RNN model. This phase begins with selecting scenarios that yield the lowest MAPE values for each normalization method. The use of MAPE as a reference in selecting the best scenarios is chosen because MAPE indicates the level of prediction error in the form of a percentage relative to the actual value. Using this

ISSN: 2720-9598

percentage makes it easier for end-users to understand. The best scenario for min-max normalization consists of the target attribute with a MAPE value of 3.9398%. Meanwhile, the best scenario for z-score normalization includes scenarios without NaN and negative attributes with a MAPE value of 10.6120%.

Figure 4 compares MAPE values between scenarios using min-max normalization and z-score normalization. The graph shows that the min-max normalization method has a MAPE value of 3.9398%, lower than the MAPE value of 10.6120% obtained from z-score normalization.

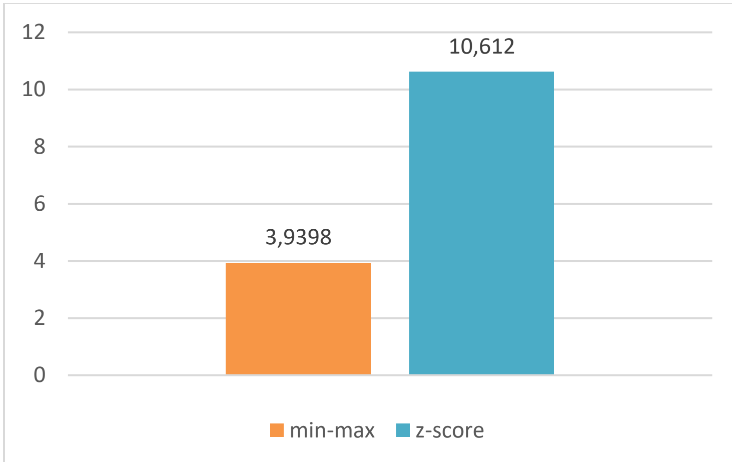


Figure 4. MAPE Comparison Results (%)

Figure 5 compares RMSE values between scenarios using min-max normalization and z-score normalization. The graph shows that the scenario with min-max normalization has an RMSE value of 0.0630, lower than the RMSE value of 0.7648 obtained from z-score normalization. Figure 6 shows that the scenario with min-max normalization has an R2 value of 0.8996, higher than the R2 value of 0.4142 obtained from z-score normalization. Previous research on comparing min-max and z-score normalization has yielded varying results. For example, one study [22] found that min-max normalization resulted in an RMSE of 9.99 compared to 10.50 for z-score. In another study [23], min-max normalization produced an MSE of 2.9079 and an MAE of 0.0040, while the z-score had an MSE of 3.0020 and MAE of 0.0130. However, in a different study [24], z-score normalization performed better with an accuracy of 0.7083 compared to min-max with an accuracy of 0.7041. These variations can be attributed to differences in data characteristics.

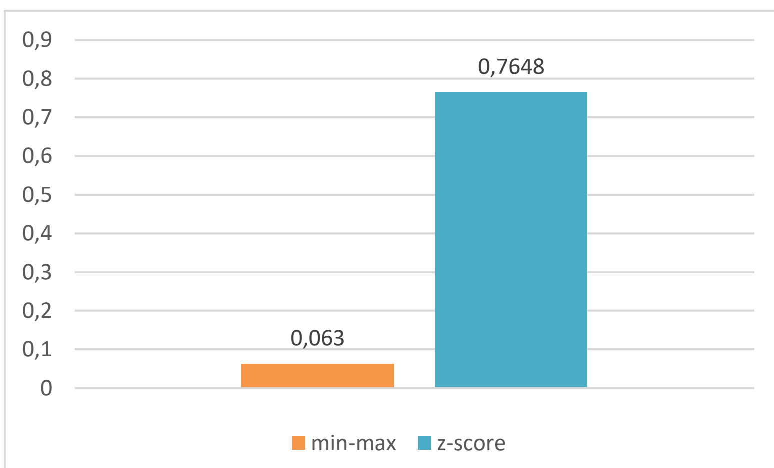


Figure 5. RMSE Comparison Results

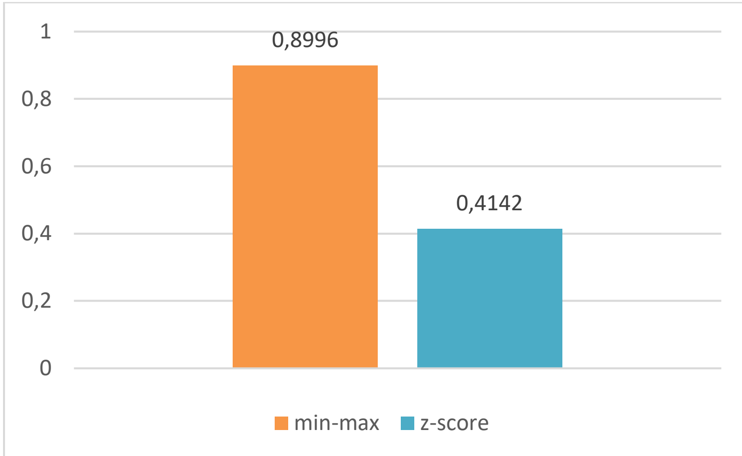


Figure 6. R2 Comparison Results

Figure 4 to Figure 6 shows that min-max normalization yields the best results. Apart from differing data characteristics, this may be due to the Tanh activation function, which passes information from -1 to 1. In other words, input neurons generated by both types of normalization are passed to the next layer. The superior performance of min-max over z-score indicates that the range of 0 to 1 significantly influences the RNN architecture in this case.
Further detailed testing of RNN for multivariate time series analysis should be conducted in the future. Recommendations for energy usage include choosing efficient and effective electronic devices and avoiding prolonged usage of appliances. Consider using alternative energy sources other than electricity during extreme weather conditions, especially for temperature regulation.

CONCLUSION

Using min-max normalization resulted in the best performance with MAPE of 3.9398%, RMSE of 0.0630, and R2 of 0.8996. Testing with z-score normalization showed slightly lower performance with MAPE of 10.6120%, RMSE of 0.7648, and R2 of 0.4142. Therefore, in comparing the two normalizations, the min-max outperformed the z-score in the case of multivariate time series data analysis. The research conducted with the RNN algorithm yielded good performance with errors below 10%. The application of attribute usage scenarios did not significantly affect the results. This is because the concept of RNN, with its repeating process and consistent weighting, allows the model to perform well.

REFERENCES

[1] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, "Recurrent Neural Networks for Multivariate Time Series with Missing Values," *Sci. Rep.*, vol. 8, no. 1, p. 6085, Apr. 2018, doi: 10.1038/s41598-018-24271-9.

[2] C. Zhang *et al.*, "A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data," in *33rd AAAI Conference on Artificial Intelligence, AAAI 2019, 31st Innovative Applications of Artificial Intelligence Conference, IAAI 2019 and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019*, 2019, pp. 1409–1416, doi: 10.1609/aaai.v33i01.33011409.

[3] F. Petropoulos and E. Spiliotis, "The Wisdom of the Data: Getting the Most Out of Univariate Time Series Forecasting," *Forecasting*, vol. 3, no. 3, pp. 478–497, Jun. 2021, doi: 10.3390/forecast3030029.

[4] A. Bagnall *et al.*, "The UEA multivariate time series classification archive, 2018," pp. 1–36, 2018, [Online]. Available: http://arxiv.org/abs/1811.00075.

[5] A. Razaque *et al.*, "Anomaly Detection Paradigm for Multivariate Time Series Data Mining for Healthcare," *Appl. Sci.*, vol. 12, no. 17, p. 8902, Sep. 2022, doi: 10.3390/app12178902.

[6] P. T. Yamak, L. Yujian, and P. K. Gadosey, "A Comparison between ARIMA, LSTM, and GRU for Time Series Forecasting," in *Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence*, Dec. 2019, pp. 49–55, doi: 10.1145/3377713.3377722.

[7] J. Feyrer, "Trade and Income—Exploiting Time Series in Geography," *Am. Econ. J. Appl. Econ.*, vol. 11, no. 4, pp. 1–35, Oct. 2019, doi: 10.1257/app.20170616.

[8] J. Runge *et al.*, "Inferring causation from time series in Earth system sciences," *Nat. Commun.*, vol. 10, no. 1, p. 2553, Jun. 2019, doi: 10.1038/s41467-019-10105-3.

[9] S.-Y. Shih, F.-K. Sun, and H. Lee, "Temporal pattern attention for multivariate time series forecasting," *Mach. Learn.*, vol. 108, no. 8–9, pp. 1421–1441, Sep. 2019, doi: 10.1007/s10994-019-05815-0.

ISSN: 2720-9598

[10] H. He, Q. Zhang, S. Bai, K. Yi, and Z. Niu, “CATN: Cross Attentive Tree-Aware Network for Multivariate Time Series Forecasting,” *Proc. AAAI Conf. Artif. Intell.*, vol. 36, no. 4, pp. 4030–4038, Jun. 2022, doi: 10.1609/aaai.v36i4.20320.
[11] J. Bernard, M. Hutter, H. Reinemuth, H. Pfeifer, C. Bors, and J. Kohlhammer, “Visual-Interactive Preprocessing of Multivariate Time Series Data,” *Comput. Graph. Forum*, vol. 38, no. 3, pp. 401–412, Jun. 2019, doi: 10.1111/cgf.13698.
[12] A. Mathew, P. Amudha, and S. Sivakumari, “Deep Learning Techniques: An Overview,” 2021, pp. 599–608.
[13] M. Akhtar and S. Moridpour, “A Review of Traffic Congestion Prediction Using Artificial Intelligence,” *J. Adv. Transp.*, vol. 2021, pp. 1–18, Jan. 2021, doi: 10.1155/2021/8878011.
[14] R. Fukuoka, H. Suzuki, T. Kitajima, A. Kuwahara, and T. Yasuno, “Wind Speed Prediction Model Using LSTM and 1D-CNN,” *J. Signal Process.*, vol. 22, no. 4, pp. 207–210, 2018, doi: 10.2299/jsp.22.207.
[15] V. Gupta, K. Sharma, and M. S. Sangwan, “Airlines passenger forecasting using LSTM based recurrent neural networks,” *Int. J. Inf. Theor. Appl.*, vol. 26, no. 2, pp. 178–187, 2019.
[16] A. Sethia and P. Raut, “Application of LSTM, GRU and ICA for stock price prediction,” in *Information and Communication Technology for Intelligent Systems: Proceedings of ICTIS 2018, Volume 2*, 2019, pp. 479–487.
[17] F. Masri, D. Saepudin, and D. Adytia, “Forecasting of Sea Level Time Series using Deep Learning RNN, LSTM, and BiLSTM, Case Study in Jakarta Bay, Indonesia,” *e-Proceeding Eng.*, vol. 7, no. 2, pp. 8544–8551, 2020.
[18] S. Huber, H. Wiemer, D. Schneider, and S. Ihlenfeldt, “DMME: Data mining methodology for engineering applications – a holistic extension to the CRISP-DM model,” *Procedia CIRP*, vol. 79, pp. 403–408, 2019, doi: 10.1016/j.procir.2019.02.106.
[19] D. M. Belete and M. D. Huchaiah, “Grid search in hyperparameter optimization of machine learning models for prediction of HIV/AIDS test results,” *Int. J. Comput. Appl.*, vol. 44, no. 9, pp. 875–886, Sep. 2022, doi: 10.1080/1206212X.2021.1974663.
[20] S. M. Nabavinejad, S. Reda, and M. Ebrahimi, “Coordinated Batching and DVFS for DNN Inference on GPU Accelerators,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 10, pp. 2496–2508, Oct. 2022, doi: 10.1109/TPDS.2022.3144614.
[21] P. Liu, Q. Yu, Z. Wu, S. Kang, H. Meng, and L. Cai, “A deep recurrent approach for acoustic-to-articulatory inversion,” in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2015, pp. 4450–4454, doi: 10.1109/ICASSP.2015.7178812.
[22] A. B. Al-Ghamdi, S. Kamel, and M. Khayyat, “Evaluation of Artificial Neural Networks Performance Using Various Normalization Methods for Water Demand Forecasting,” *Proc. - 2021 IEEE 4th Natl. Comput. Coll. Conf. NCCC 2021*, 2021, doi: 10.1109/NCCC49330.2021.9428856.
[23] S. Bhanja and A. Das, “Impact of Data Normalization on Deep Neural Network for Time Series Forecasting,” pp. 5–10, 2018, [Online]. Available: http://arxiv.org/abs/1812.05519.
[24] J. Paparrizos, C. Liu, A. J. Elmore, and M. J. Franklin, “Debunking Four Long-Standing Misconceptions of Time-Series Distance Measures,” in *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, Jun. 2020, pp. 1887–1905, doi: 10.1145/3318464.3389760.